HEQUATE IN ACTION

Compiler Team - Quantum Computing Methods & Implementation Group (QMI)

Software and research to unlock the full potential of DLR quantum computers.

Dr. David da Costa, Institute for Software Technology

The compiler Team

Who we are

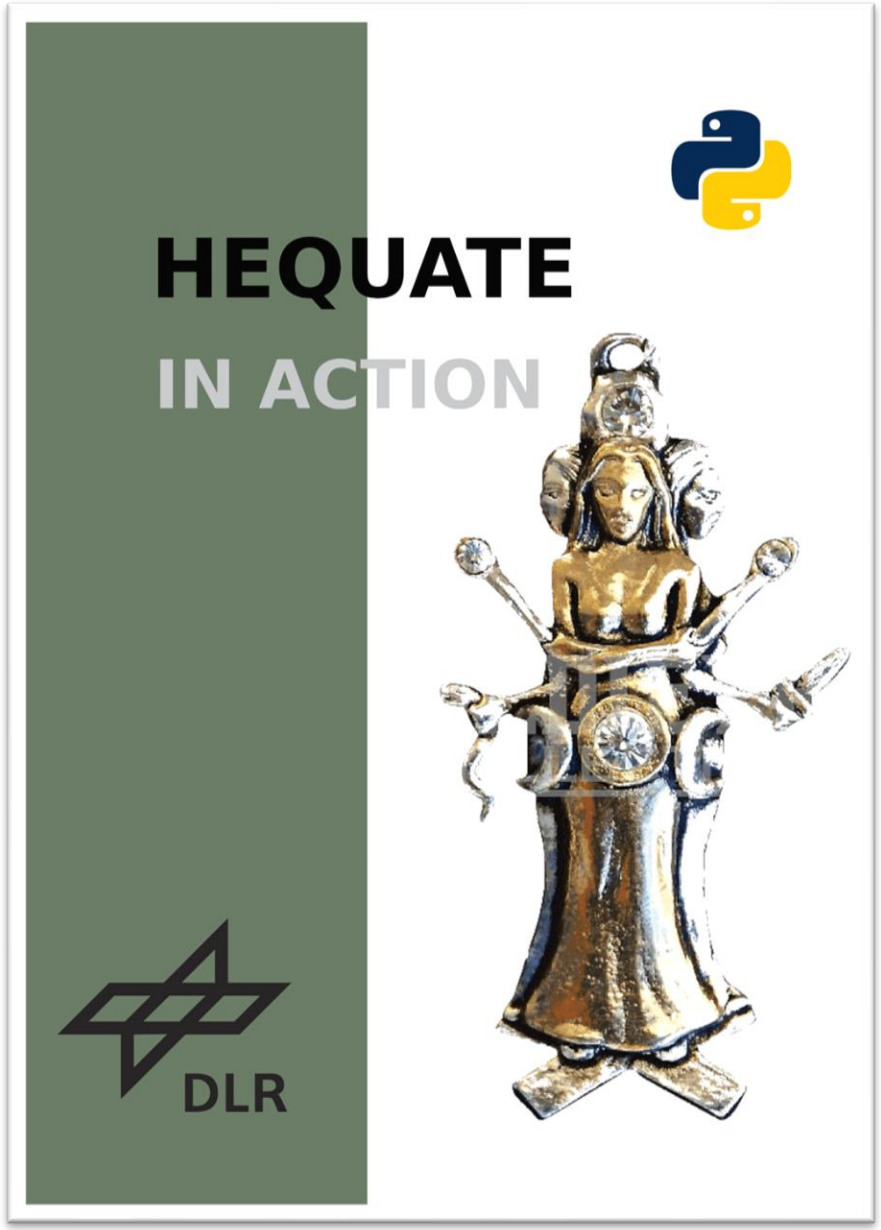
- David da Costa
- Monika Das
- Thomas Keitzl
- Johannes Renkl
- Thomas Stehle

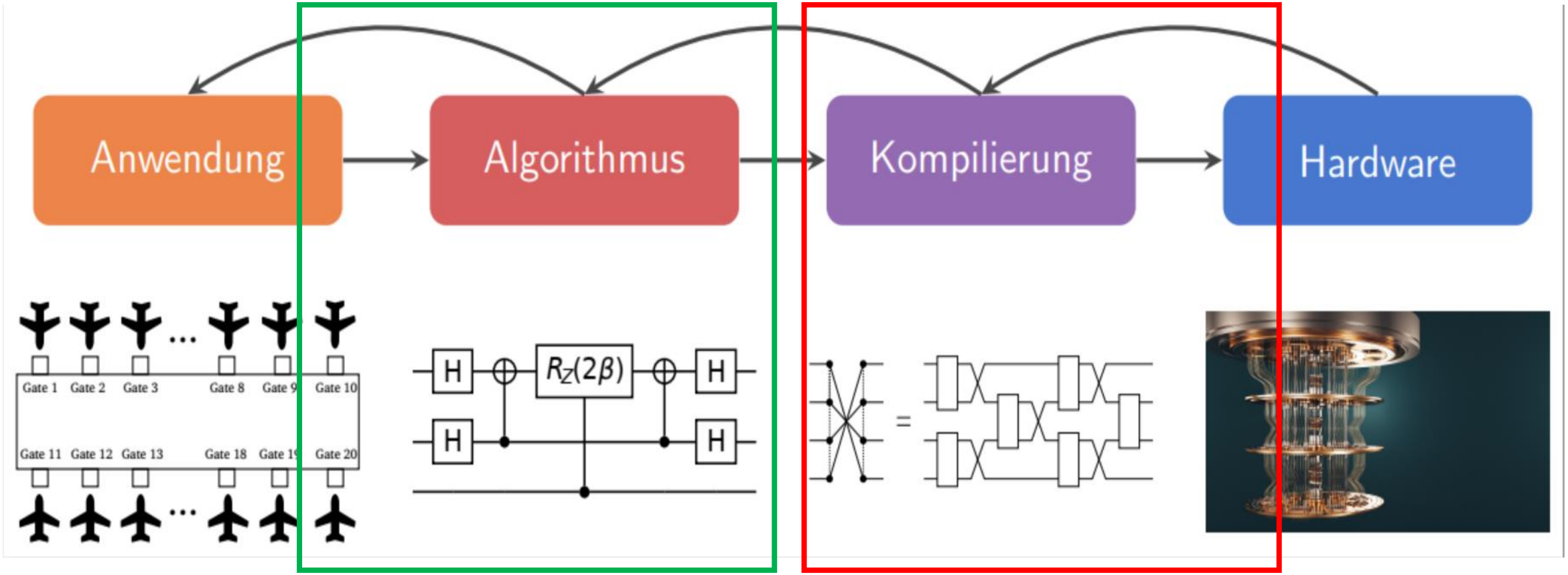
What our Mission is

- Compiler steps
  - Quantum error correction & mitigation
  - Compilation (Hardware-specific)
  - Circuit optimization
  - Hybrid (classical & quantum) computing

- <u>Make the efforts to bring a Algorithms to the machine</u>

- DLR projects
  - ALQU/CLIQUE: Compilation, error correction, integration




Anwendung
Algorithmus
Kompilierung
Hardware
Gate 1
Gate 2
Gate 3
Gate 8
Gate 9
Gate 10
Gate 11
Gate 12
Gate 13
Gate 18
Gate 19
Gate 20
H
H
$R_Z(2\beta)$
H
H
=

Research Activities: Software Stack and QC Integration

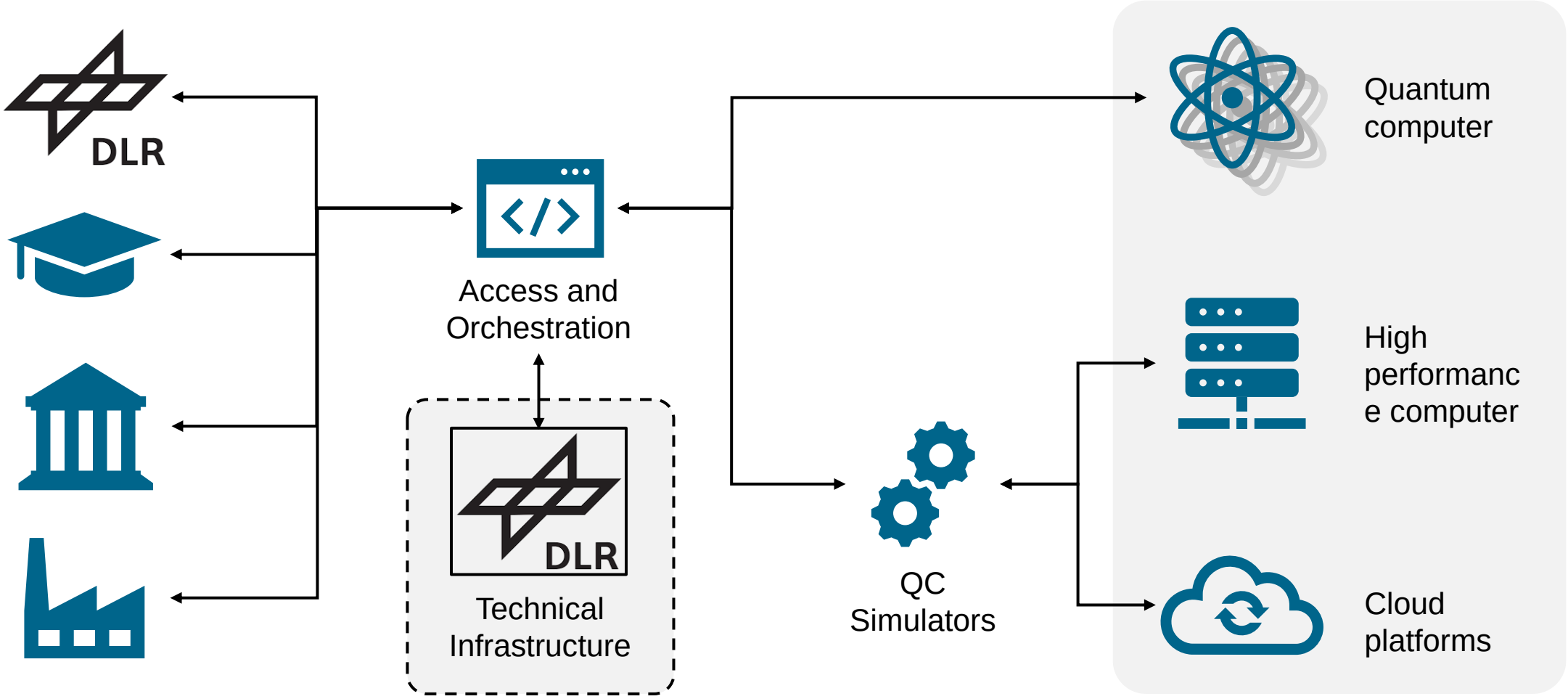
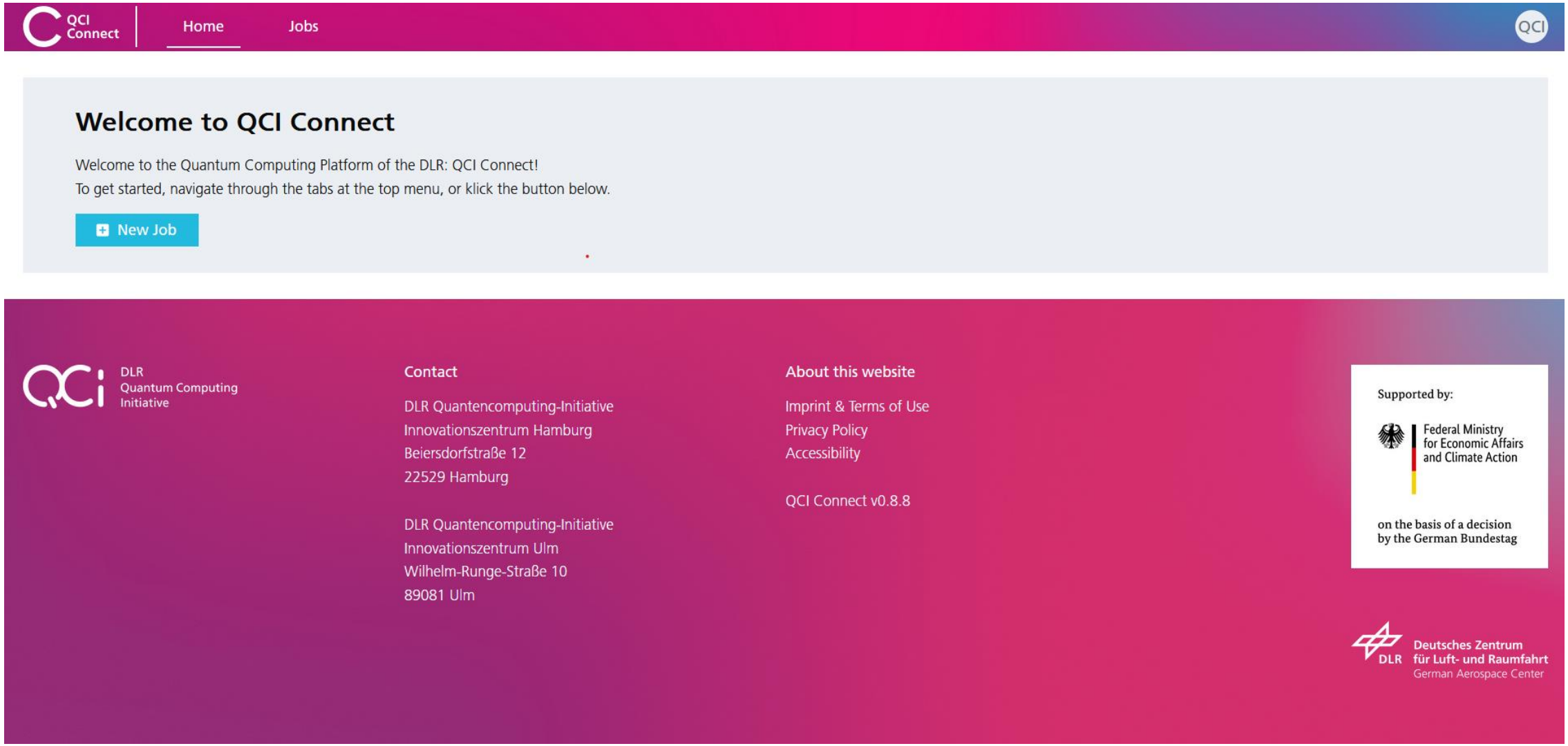
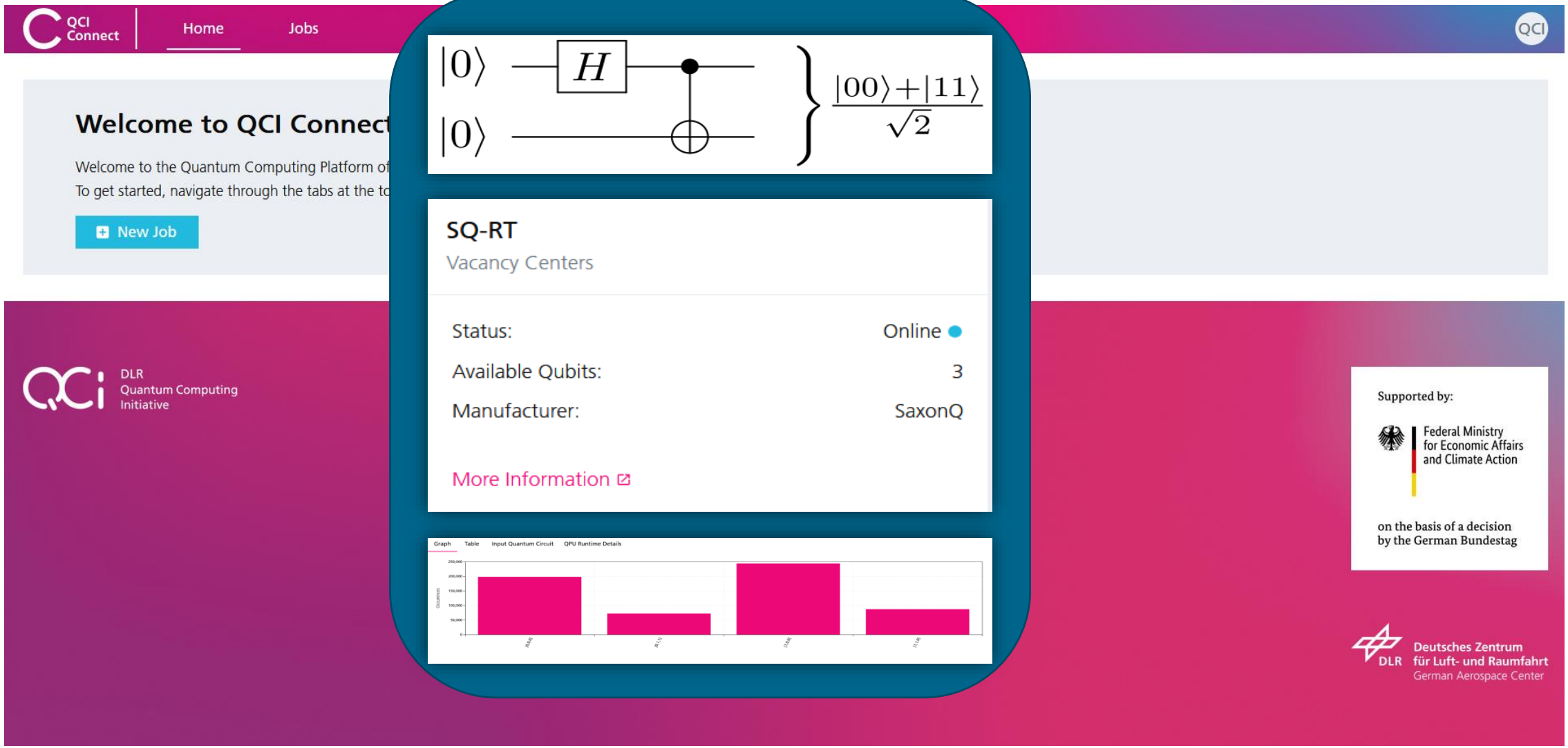


Image source: https://qci.dlr.de/posterpraesentationen/

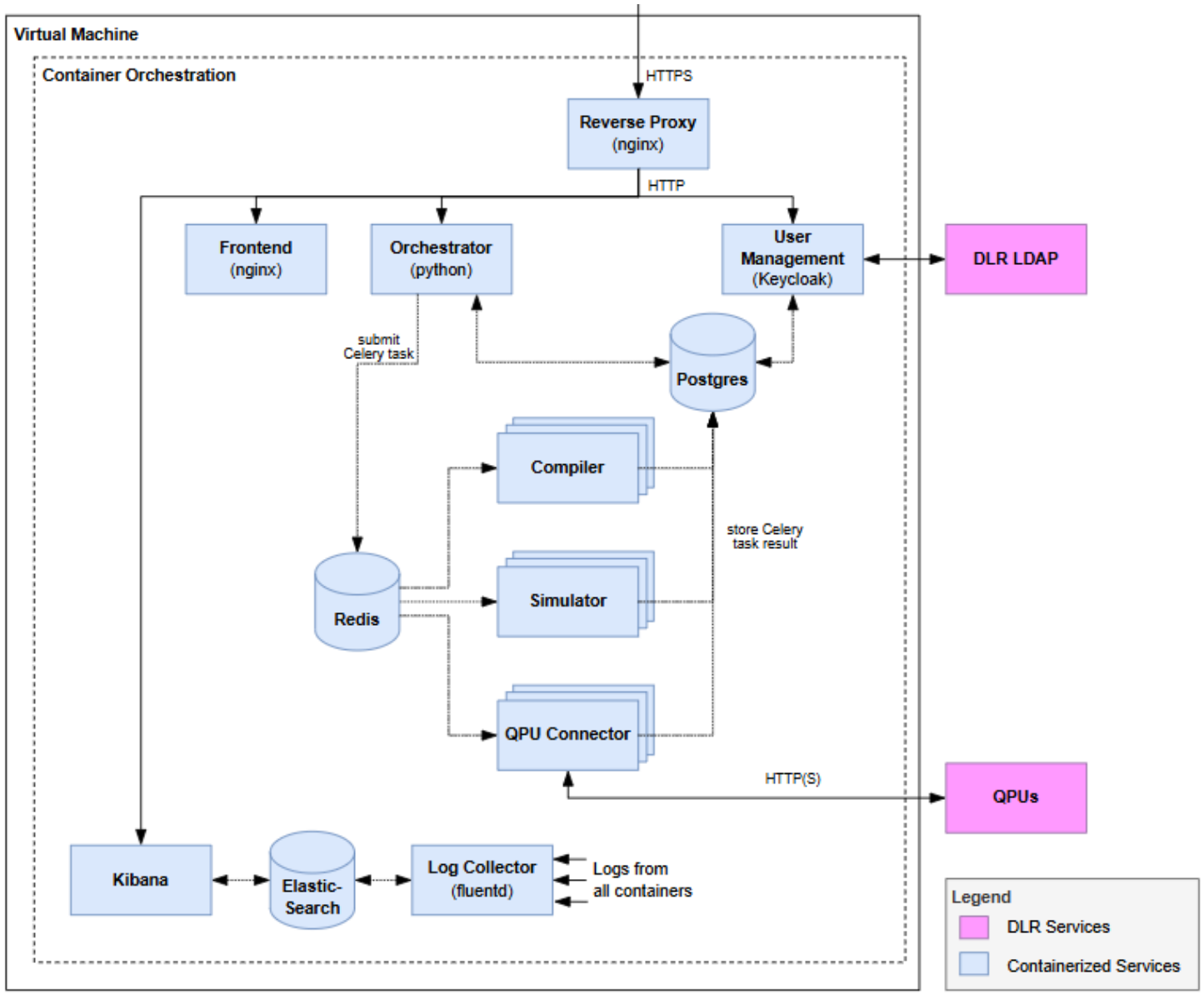
Find at http://qc-plat-[pro|sta|dev].dlr.de/Home (VPN-SC)



Find at http://qc-plat-sta.dlr.de/Home (VPN-SC)



The QCI Connect



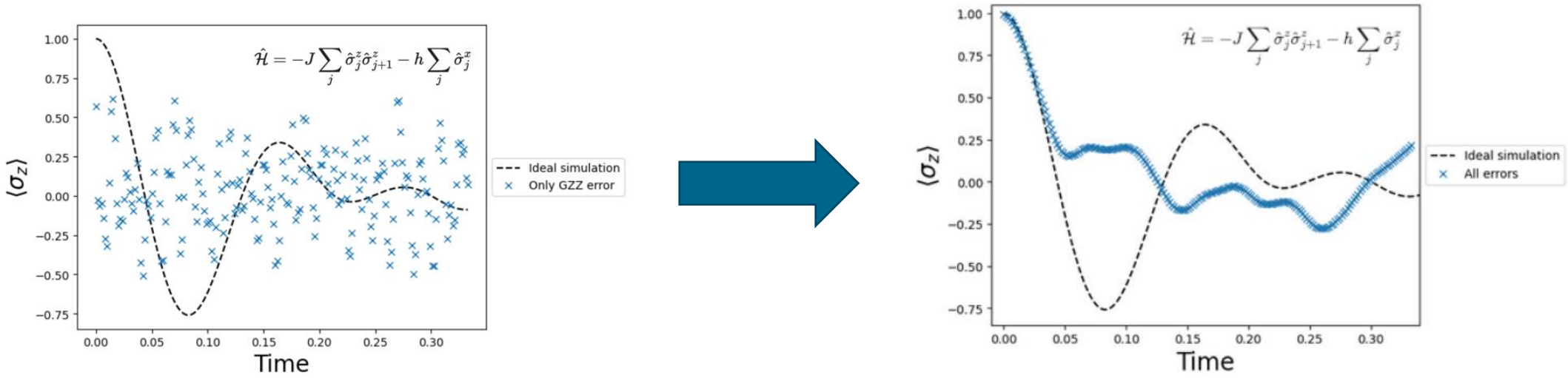
HEQUATE

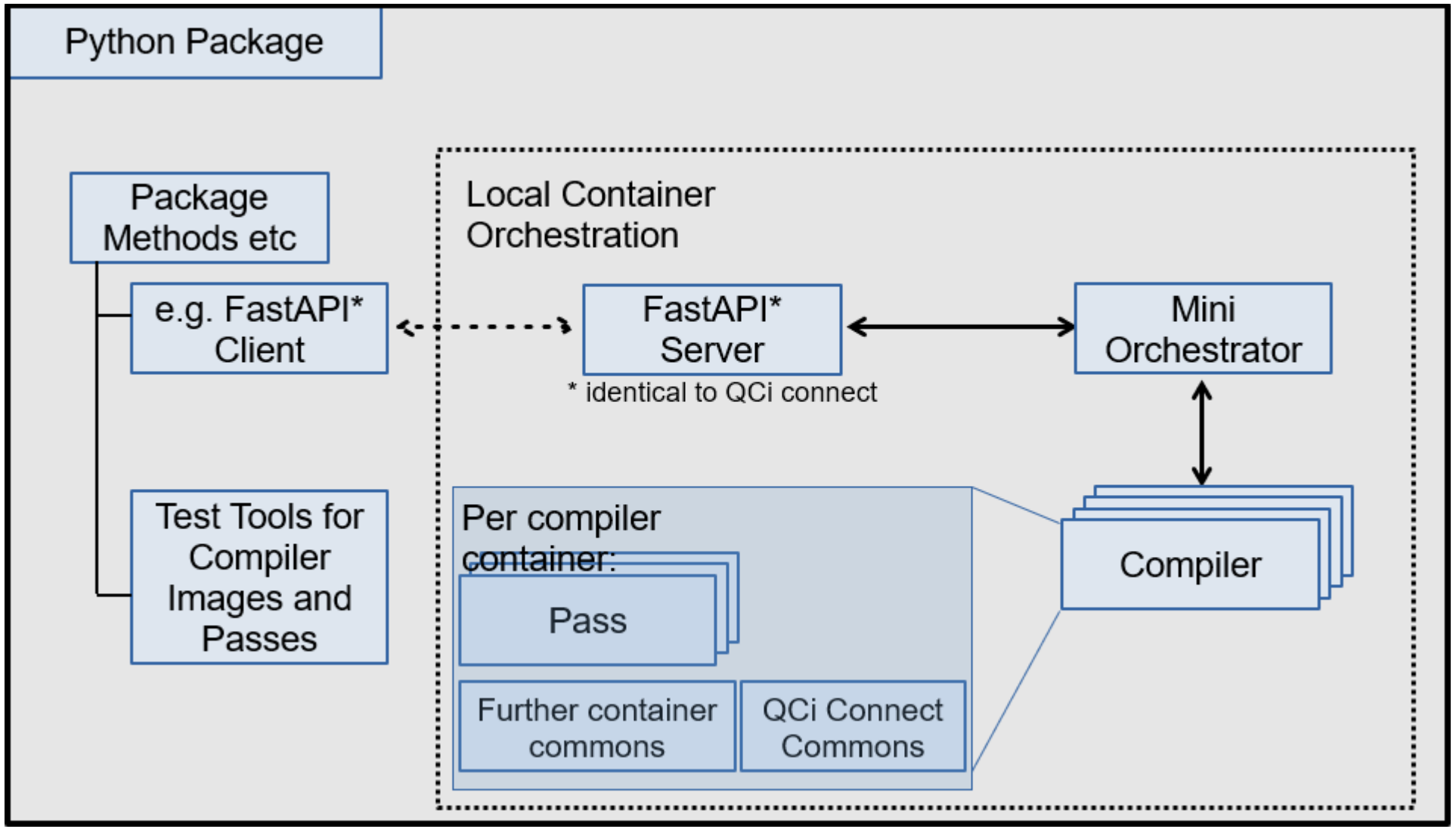
IN ACTION

DLR

Our Motivation

- The possibility to develop local tools (compiler steps)
  - Error mitigation: Zero noise extrapolation, Pauli Twirling
  - Quantum Signal Processing (reimplements PITE without mid-circuit measurements)
- Allow for more complicate setup without compromise security
- Private methods that are not yet public / open source
- Add new components, like simulators or other compilers
- Pre- and/or Post-processing (ZNE - combine different intermediate circuits)
- Allowed the tools to be provided in other programming languages (c++, Julia)





At the moment circuits are exchange through **OPENQASM**.

Compiler Pass Definition Example

YAML

```yaml
name: "QSP"
description: "Quantum Signal Processes the given block encoding according to provided polynomial."
compiler_name: "pyqsp"
compiler_alias: "pyqsp"
compiler_id: 0
Options:
  - name: polynomial
    description: "Polynomial to apply to block encoded operator."
    schema:
      type: list[int]
```

Time for some demos…